A proposed Algorithm for Interactive Geometric Shapes Recognition

Dr. Hayder Hadi Abbas
Software Engineering Department, Al-Mansour University College/Baghdad
Email: hayder19722003@yahoo.com

Received on: 19/5/2013 & Accepted on:9/1/2014

ABSTRACT

A geometric shapes recognition algorithm is proposed in this paper. This algorithm is interactive which means that if not all the desired shapes are recognized then the recognition process is repeated by the user to improve the capability of the process.

The user can change threshold value when converting the image from gray to binary many times during one program execution to get the required results.

The proposed algorithm can select the desired shape individually even if there are different shapes in the image (multi-shapes image).

This algorithm is software implemented using Matlab programming language and then implemented program is used to recognize any of the following shapes (the shapes must be distinct and not overlapped):

Seven types of triangle; pentagons, hexagons, squares, rectangles, circles, ellipse shapes, rhombus, and parallelogram.

The recognition capability of the implemented software is tested for different cases and from these cases many points are concluded.

Keywords: Basic geometric shapes, interactive, recognition, Matlab programming language.

خوارزمية مقترحة للتمييز بين الأشكال الهندسية بطريقة تفاعلية

الخلاصة :

تـم فـي هـذا البحـث أقتـراح خوارزميـة للتمييـز بـين الأشـكال الهندسـية فـي صـورة معينـة. أن الخوارزمية المقترحة تفاعلية وهذا يعني أنه في حالـة عدم تمييز كل الأشكال المطلوبـة فـأن مستخدم الخوارزمية يمكن أن يكرر عملية التمييز بين الأشكال الهندسية لكي يحسن من قابلية عملية التمييز. أن المستخدم يستطيع تغيير حد العتبـة (threshold value) عند تحويل الصـورة من النـوع الرمـادي (gray) الى النوع الثنائي (binary) ولعدد من المرات أثناء عملية تنفيذ واحدة ومستمرة للبرنـامج ولغاية الحصول على النتائج المطلوبة.

أن الخوارزومية المقترحة تستطيع تمييز أي شكل هندسي وبشكل منفصل حتى وأن كانت الصورة المدخلة للخوارزمية فيها أشكال هندسية مختلفة (صورة متعددة الأشكال).

تم بناء الخوارزمية المقترحة برمجياً بواسطة الحزمة البرمجية (Matlab). أن البرنامج الحاسوبي يستطيع تمييز أي شكل من الأشكال التالية (يجب أن تكون الأشكال منفصلة وغير متداخلة مع بعضها):
سبع أنواع من المثلثات، الأشكال الخماسية، الأشكال السداسية، المربعات، المستطيلات، الدوائر، الأشكال البيضوية، المعين و متوازي المستطيلات.
أن قدرة التمييز للبرنامج تم اختبارها لحالات مختلفة ومن هذه الحالات تم أستنتاج عدة نقاط.

INTRODUCTION

Recognition of geometric shapes has garnered new attention with the widespread adoption of Personal Digital Assistants (PDAs). A geometric shape is the geometric information which remains when location, scale, orientation and reflection are removed from the description of a geometric object. That is, the result of moving a shape around, enlarging it, rotating it, or reflecting it in a mirror is the same shape as the original, and not a distinct shape. Many two-dimensional geometric shapes can be defined by a set of points or vertices and lines connecting the points in a closed chain, as well as the resulting interior points. Such shapes are called polygons and include triangles, squares, and pentagons. Other shapes may be bounded by curves such as the circle or the ellipse.

While conventional off-line approaches to recognition of geometric shapes have long focused on raw classification performance, on-line systems raise a different set of issues. In off-line approaches, geometric shapes in CAD drawings are input precisely either by human drafts-people or through computer peripherals and recognizing these shapes deal mainly with noise introduced by sources outside the process, such as copy degradation or poor photographic reproduction. In online applications however, the noise is inherent to the process of information gathering and shapes are often sketched poorly due to media, operator and process limitations, yielding imperfect and ambiguous shapes that even humans find difficult to distinguish. One further difference from off-line algorithms is that input data are sequence of points rather than image bitmaps [1, 2, 3].

The idea of pen-centric interfaces is not new, even if the fact that sketches and planar diagrams as a way of communication precede the invention of writing by more than 30 centuries is ignored. In 1963, Sutherland presented Sketch-pad which is the first interactive system that used one light pen to draw diagrams directly over the screen surface. The main limitation of this system resided in the recognition capabilities, limited resources and high cost of the computer used [1].

Nevertheless, due in part to ergonomic problems with the light pen and the invention of the mouse in 1964, present-day graphical interfaces relegated pen based systems to specific CAD applications until the appearance of the first pen computers in 1991. Tappert provides a comprehensive survey of the work developed in the area of non-interactive graphics recognition for the entering data and diagrams as vectorial information in CAD systems [2].

Mark Gross describes a system fundamentally based in sketches that are partially interpreted for use in architecture drawings using a recognizer substantially simpler and less robust than ours [3].

The Newton system is one of the first hand held pen based computers which incorporates a hand written recognizer, a shape recognizer and a gesture recognizer. The shape recognizer only recognizes solid shapes and requires the shapes to be drawn with just one stroke and aligned with the axes. The gesture recognizer is more suitable for text editing than for schematic drawing [3].

Sanket Rege, Rajendra Memane, Mihir Phatak, and Parag Agarwal present an approach involving digital image processing and geometric logic for recognition of two dimensional shapes of objects such as squares, circles, rectangles and triangles as well as the color of the object [4].

Even though significant progresses were made, most notably in the past decade, more work is required to make handwriting and sketch recognition systems truly usable in most settings.

In this paper an algorithm is proposed to extract geometric shapes features and then use the extracted features to distinguish between these shapes.

THE BASIC GEOMETRIC SHAPES

In this section the basic geometric shapes such as triangles, quadrilaterals, polygons, circles, and ellipse are discussed.

Triangle [5, 6]

A closed figure consisting of three line segments linked end-to-end. The triangle properties are:

1. Median

The median of a triangle is a line from a vertex to the midpoint of the opposite side. The three medians intersect at a single point, called the centroid of the triangle.

2. Area

The area of a triangle is given by the:

$$\text{Area}= \frac{b \times a}{2} \qquad \ldots(1)$$

where

b is the length of the base, and a is the length of the corresponding altitude.

3. Perimeter

It is distance around the triangle or the sum of its sides. Figure (1) is a graph of a triangle.

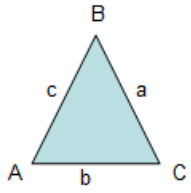

Figure (1) Graph of a triangle.

There are seven types of triangle:

1. Isosceles  in which two sides are equal.

2. Equilateral  in which all sides are equal.

3. Scalene  in which no sides are equal.

4. Right Triangle  in which one angle = 90°.

5. Obtuse  in which one angle is greater than 90°.

6. Acute  in which all angles are less than 90°.

7. Equiangular  in which all interior angles are equal.

Quadrilateral [7, 8]

A quadrilateral is a closed figure with exactly four line segments (or sides). Four Types of quadrilateral are used in this paper and are discussed here.

1) Rectangle

Rectangle is quadrilateral that has four right angles. The rectangle properties are:

1. Opposite sides are parallel and congruent.
2. The diagonals bisect each other.
3. The diagonals are congruent.

The area of a rectangle is given by the:

$$\text{Area} = w \times h \qquad \ldots(2)$$

where

w is the width, and h is the height.

The perimeter of a rectangle is given by the:

$$\text{Perimeter} = 2 \times (w + h) \qquad \ldots(3)$$

Figure. 2 is a graph of a rectangle.

Figure (2) Graph of a rectangle.

2) Square

A square is a rectangle that has four congruent sides. The square properties are:

1. Perimeter

It is the distance around the square. All four sides are by definition the same length, so the perimeter is four times the length of one side, or:

$$\text{Perimeter} = 4s \quad \ldots (4)$$

where

s is the length of one side.

2. Area

Like most quadrilaterals, the area is the length of one side times the perpendicular height. So in a square this is simply:

$$\text{Area} = s^2 \quad \ldots (5)$$

Figure (3) is a graph of a square.

Figure (3) Graph of a square.

3) Parallelogram

A quadrilateral with both pairs of opposite sides parallel. The parallelogram properties are:

1. Base

Any side can be considered a base. If used to calculate the area the corresponding altitude must be used.

2. Altitude (height)

The altitude (or height) of a parallelogram is the perpendicular distance from the base to the opposite side (which may have to be extended).

3. Area

The area of a parallelogram can be found by multiplying a base by the corresponding altitude.

4. Perimeter

It is the distance around the parallelogram or the sum of its sides.

$$\text{Perimeter} = 2 \times (w + h) \quad \ldots(6)$$

Where

w is the base length of the parallelogram, and h is the side length. Figure (4) is a graph of a parallelogram.

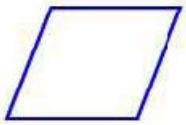

Figure (4) Graph of a parallelogram.

4) Rhombus

Rhombus is a quadrilateral with all four sides equal in length. A rhombus is actually just a special type of parallelogram. Recall that in a parallelogram each pair of opposite sides are equal in length. With a rhombus, all four sides are the same length. It therefore has all the properties of a parallelogram.

Figure (5) is a graph of a rhombus.

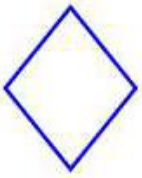

Figure (5) Graph of a rhombus.

Polygons [9, 10, 11]

It is a number of coplanar line segments, each connected end to end to form a closed shape. Polygons are one of the most all-encompassing shapes in geometry. Below are the types of polygons.

1. Regular : It is a polygon with all sides and interior angles are the same.

2. Irregular: In this case each side may be a different length, each angle may be a different measure.

3. Convex: In this case all interior angles are less than 180°, and all vertices point outwards away from the interior.

4. Concave: In this case one or more interior angles are greater than 180°. Some vertices push 'inwards' towards the interior of the polygon.

5. Self-intersecting or Crossed: It is a polygon where one or more sides crosses back over another side, creating multiple smaller polygons. Most of the properties and theorems concerning polygons do not apply to this shape. The polygons properties are:

1. Area

For regular polygons the area can be calculated as given below:

$$area = \frac{s^2 N}{4\tan\left(\frac{180}{N}\right)} \qquad \ldots(7)$$

where

S is the length of any side, and N is the number of sides.

For irregular polygons things are a little harder since there no general formulae.

2. Perimeter

The distance around a polygon. The sum of its side lengths.

$$\text{Perimeter} = ns \qquad \ldots(8)$$

where

n is the number of sides, s is the length of any side.
Many polygons have names based on the number of sides. A 5-sided polygon is called a pentagon for example. Beyond about 10 sides, most people call them an "n-gon". For example a 15-gon has 15 sides. This seems easier to remember and understand. However, there are some names that do occur in everyday experience:
triangle with 3 sides, quadrilateral with 4 sides, tetragon with 4 sides, pentagon with 5 sides, hexagon with 6 sides, heptagon with 7 sides, octagon with 8 sides, decagon with 10 sides, and dodecagon with 12 sides.
In this paper, two polygons are considered which are pentagon and hexagon as shown in Figure (6).

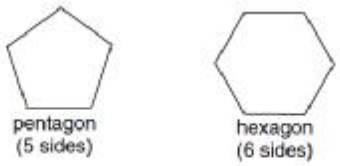

Figure (6) Graph of two types of polygons.

Circle [12, 13]
A line forming a closed loop, every point on which is a fixed distance from a center point.
The circle properties are:
1. Center
A point inside the circle. All points on the circle are equidistant (same distance) from the center point.
2. Radius
The radius is the distance from the center to any point on the circle. It is half the diameter.
3. Circumference
The circumference is the distance around the circle. It is calculated as given below:

$$\text{Circumference} = 2\pi R \qquad \ldots(9)$$

Where
R is the radius of the circle, and π is Pi, approximately 3.142.
4. Area
Strictly speaking a circle is a line, and so has no area. What is usually meant is the area of the region enclosed by the circle. It is calculated as given below:

$$area = \pi R^2 \qquad \ldots (10)$$

The graph of a circle is shown in Figure (7).

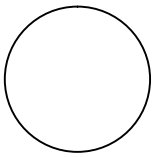

Figure (7) Graph of a circle.

Ellipse [12, 13]

A curved line forming a closed loop, where the sum of the distances from two points (foci) to every point on the line is constant.

An ellipse is defined by two points, each called a focus. If you take any point on the ellipse, the sum of the distances to the focus points is constant. The size of the ellipse is determined by the sum of these two distances. The sum of these distances is equal to the length of the major axis (the longest diameter of the ellipse).

If the foci are at the same location, the ellipse is a circle. A circle is, in fact, a special case of an ellipse.

The properties of the ellipse are:

1. Center

A point inside the ellipse which is the midpoint of the line segment linking the two foci. The intersection of the major and minor axes.

2. Major / minor axis

The longest and shortest diameters of an ellipse.

3. Foci (Focus points)

The two points that define the ellipse.

4. Perimeter (circumference)

The perimeter is the distance around the ellipse and it is not easy to calculate.

5. Area

The number of square units it takes to fill the region inside an ellipse.

$$area = \pi jn \qquad \dots (11)$$

where

j is the major radius of the ellipse, and n is the minor radius of the ellipse.

In an ellipse, if you make the major and minor axis the same length, the result is a circle, with both foci at the center. The graph of an ellipse is shown in Figure (8).

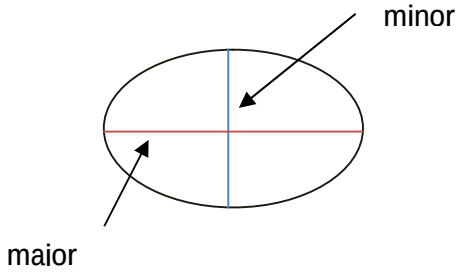

Figure (8) Graph of an ellipse.

EXTRACTING THE PROPERTIES OF THE SHAPES [9, 14]

The first task to be accomplished is to extract the properties of the geometric shapes to be able to recognize among them. Since the proposed software is implemented using Matlab then the properties of the shapes is extracted using the function 'regionprops'. The function 'regionprops' gives many properties. Some of these properties are discussed below:

1. The area of each shape which is the sum of the pixels within shapes.

2. Centroid of each shape in the image which is a point or pixel of the image coordinates (x, y) which is the shapes' center of gravity. For example centroid of the circle is the center of the circle and centroid of the square is the meeting point of axes.
3. 'BoundingBox' of the shape in the image which is the smallest rectangle holding the shape which gives the characteristic point coordinates of leftest upper from 'BoundingBox' and the length and width of the shape [(x,y), L and W]. 'BoundingBox' is useful in many applications, including shapes and Arabic and English characters recognition.
4. Extent of the shape in the image which is the ratio of the area of each shape to 'Boundingbox' shape area.

Extent = Area (object) / Area ('Boundingbox')

5. 'Extrema' of the shape in the image. These eight feature points are:
Top-left, top-right, right-top, right-bottom, bottom-right, bottom-left, left-bottom, and left-top as shown in Figure (9):

Figure (9) 'Extrema' of the shape.

6. The 'MajorAxisLength' of the shape which gives the length of the major axis of the oval containing the shape as shown in Figure. 10:

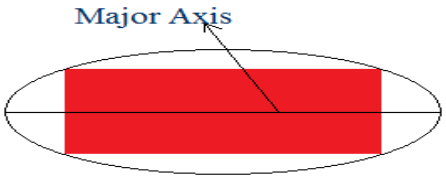

Figure (10) 'MajorAxisLength' of the shape.

7. The 'MinorAxisLength' of the shape which gives the length of the minor axis of the oval containing the shape as shown in Figure. 11:

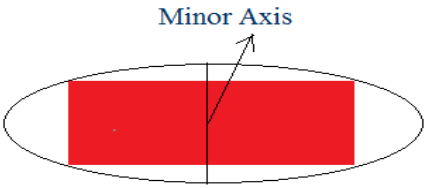

Figure (11) 'MinorAxisLength' of the shape.

THE PROPOSED ALGORITHM IMPLEMENTATION

The proposed algorithm can perform geometric shapes recognition process according to the following steps:

1. Read the input image (RGB).
2. Convert the input image to GRAY image.
3. Convert the GRAY image to BINARY image.
4. Extract the properties of the shape using 'regionprops' function.
5. Chose the shape to be recognized.
6. For all shapes in the image perform the following tasks:

a. Compare between top-left point of the shape and top-right point. If the two points are equal then the shape may be triangle, pentagon, or hexagon.

b. Compare between right-top point of the shape and right-bottom point. If the two points are equal then the shape may be triangle or pentagon. Otherwise, the shape is hexagon.

c. Compare between bottom-right point of the shape and right-bottom point. If the two points are equal then the shape is triangle. Otherwise, compare between bottom-right point of the shape and bottom-left point. If the two points are not equal then the shape is pentagon. Otherwise, it may be rhombus or parallelogram. In this case, another parameter must be examined:

1. If 'Length' of the shape is equal to 'Width' of the shape and 'Extent' is very close to 0.5 then the shape is rhombus.
2. Otherwise, the shape is parallelogram.

d. If steps (a), (b), and (c) are not met then compare top-right point of the shape and right-top point.

1. If the two points are equal then compare 'Length' of the shape with 'Width' of the shape and check if 'Extent' is very close to 1. If the last condition is true then the shape is square. Otherwise, the shape is rectangle.
2. If the two points are not equal then compare 'MajorAxisLength' with 'MinorAxis Length' and 'Length' of the shape with 'Width' of the shape. If the last condition is true then the shape is circle. Otherwise, the shape is ellipse.

7. After checking all the shapes in the image, display the output image with the shape to be recognized is selected.
8. If the recognition process is incorrect (some of the shapes to be recognized are not selected) then repeat the steps 6 and 7.

THE PROPOSED ALGORITHM TESTING

The proposed algorithm is tested for different cases to prove its recognition capability. These cases are:

Testing the Recognition Capability in Multi-Shapes Image

In this case an image that contains many geometrical shapes is applied to the implemented program as shown in Figure (12). The results of shapes recognition process for the nine geometrical shapes given before are shown in the Figure (13) to Figure (21) (the recognized shape is assigned using the sign 'x').

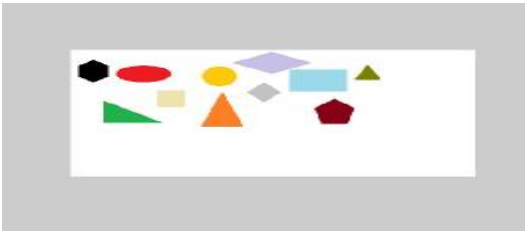

Figure (12) The Input image.

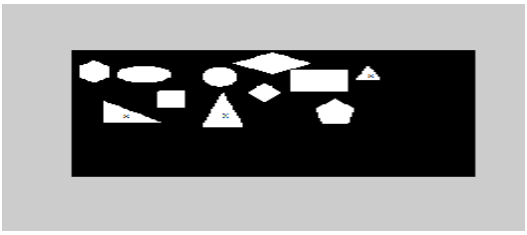

Figure (13) Triangle shapes recognition.

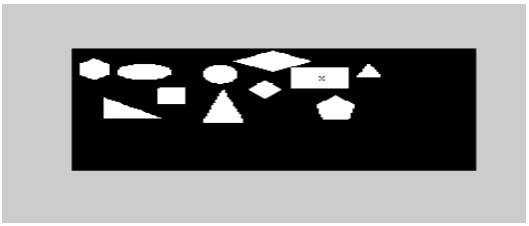

Figure (14) Rectangle shape recognition.

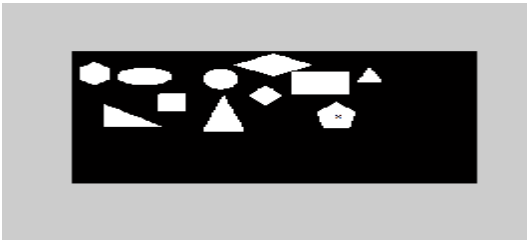

Figure (15) Pentagon shape recognition.

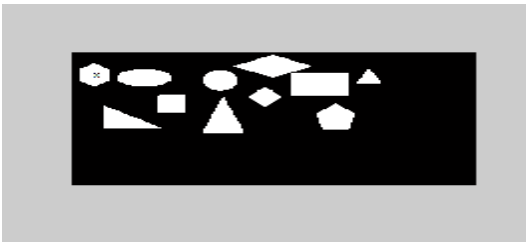

Figure (16) Hexagon shape recognition.

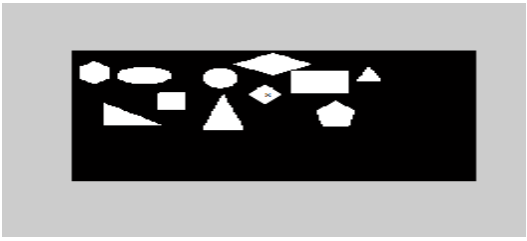

Figure (17) Rhombus shape recognition.

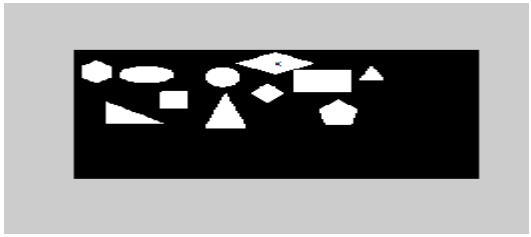

Figure (18) Parallelogram shape recognition.

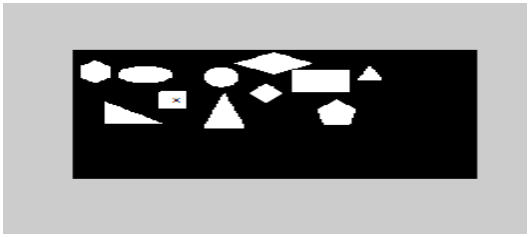

Figure (19) Square shape recognition.

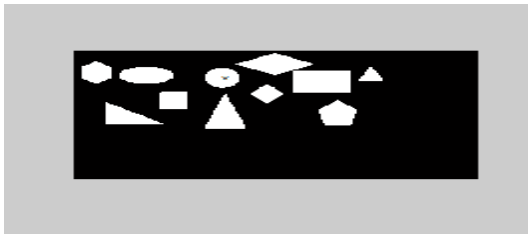

Figure (20) Circle shape recognition.

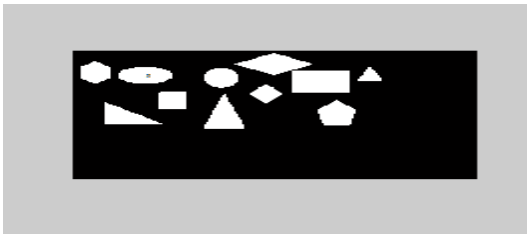

Figure (21) Ellipse shape recognition.

Testing the Interactivity of the Proposed Algorithm

In this case the capability of the implemented program to deal with incomplete shapes recognition process (some of the shapes in the image are not recognized) is investigated. In this case, the threshold value when converting the image from gray to binary is used to control the recognition process until the user is satisfied. By increasing the threshold value during program execution, the number of shapes in the image will decrease and then the recognition capability will decrease. On the contrary, decreasing the threshold value during program execution lead to an increase in the number shapes and accordingly increasing the recognition capability. Thus, the implemented program is interactive because the user can increase or decrease the threshold value many times (during one program execution) to recognize the desired shapes. The results for ellipse shapes recognition in a real image 'coins.png' Figure(22) are shown in Figure (23) to Figure (25). These results are obtained in one execution and by changing the threshold value as necessary.

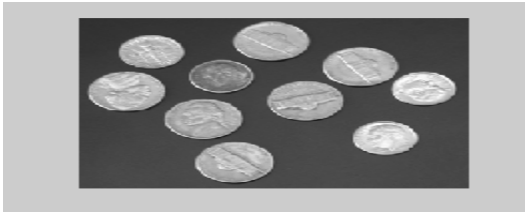

Figure (22) The input image.

Figure (23) Recognition process for threshold value = 0.7.

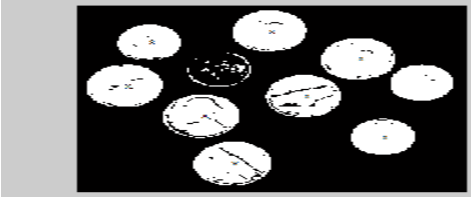

Figure (24) Recognition process for threshold value = 0.6.

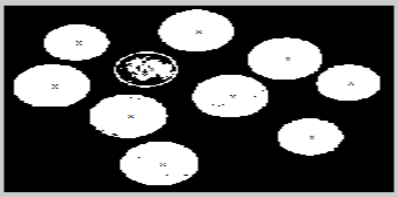

Figure (25) Recognition process for threshold value = 0.5.

CONCLUSIONS

In this paper, a computer program is implemented using Matlab. Then this program is used to recognize any of the following shapes (the shapes must be distinct and not overlapped):

Seven types of triangles; pentagons, hexagons, squares, rectangles, circles, ellipse shapes, rhombus, and parallelogram.

The conclusions from this work are:

1. The program can recognize each shape individually even if there are different shapes in the image (multi-shapes image) as shown in Figure (12).
2. This program accomplishes a huge detection process since it can recognize round shapes circle Figure (20) and ellipse Figure (21), quadrilateral (square Figure (19), rectangle Figure (14), rhombus Figure (17) and parallelogram Figure (18) and polygons triangle Figure (13), pentagon Figure (15) and hexagon Figure (16) even if they are in the same image.
3. Any of the seven types of triangles can be recognized by the implemented program.
4. The program is interactive due to the fact that if some of the desired shapes are not recognized then the user can change the threshold value during program execution and the whole recognition process (section 4) is repeated again. The interactivity in this case is the capability of the user to change the threshold value many times during one program execution to get satisfactory results (as shown in Figure 23, Figure 24, and Figure 25).

REFERENCES

[1] L. Schomaker, “From handwriting analysis to pen-computer applications”, Electronics and Communication Engineering Journal, June 1998.

[2] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara, “The state of the art in on-line handwriting recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(8):787–807, August 1990.

[3] Joaquim A. Jorge and Manuel J. Fonseca, “ A simple approach to recognize geometric shapes interactively”, Proceeding GREC '99 Selected Papers from the Third International Workshop on Graphics Recognition, Recent Advances Pages 266-276, Springer-Verlag, London, UK, 2000.

[4] Sanket Rege, Rajendra Memane, Mihir Phatak, and Parag Agarwal, “2D geometric shape and color recognition using digital image processing”, International Journal of

Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 2, Issue 6, June 2013.
[5] D. Acheson, 1089 AND ALL THAT: A JOURNEY INTO MATHEMATICS, Oxford University Press, Oxford, 2002.
[6] Young, Cynthia Y, PRECALCULUS, John Wiley and Sons, p. 831, ISBN 0-471-75684-9, 2010.
[7] Zalman Usiskin and Jennifer Griffin, "The classification of quadrilaterals, a study of definition", Information Age Publishing, 2008, p. 59, ISBN 1-59311-695-0.
[8] Mitchell, Douglas W., "The area of a quadrilateral", Mathematical Gazette, July 2009.
[9] Dugopolski, Mark, COLLEGE ALGEBRA AND TRIGONOMETRY, 4th edition, Addison Wesley, 2006.
[10] Altshiller-Court, Nathan, COLLEGE GEOMETRY, Dover, 2007.
[11] Cartensen, Jens, "About hexagons", Mathematical Spectrum 33(2), (2000-2001), 37-40.
[12] Gibson C. G., "Elementary geometry of differentiable curves: an undergraduate introduction", Cambridge University Press, 2001, p. 127, ISBN 9780521011075.
[13] Watkins, Matthew, "Useful Mathematical and Physical Formulae", Walker and Co., 2000.
[14] Matlab help, available at
: http://www.mathworks.com/help/images/ref/regionprops.